```
#-***********************
#
#
#       -*** UCL's default local entry ***-
#       -*** this is different for every PP site   ***-
#
# (Folded for readabilty)
#
default:lpsap='"591"/Internet=bells',
     lmta="UK.AC.UCL.CS",lpass=" "
#
#       -*** NASA ***-
#
"591"/Internet=titan.arc.nasa.gov:rname="us-nasa",
    rmta="titan.arc.nasa.gov",rpass=" "
"591"/Internet=128.102.128.5:rname="us-nasa",
    rmta="titan.arc.nasa.gov",rpass=" "
#
# -*** UCL ***-
#
"591"/X121+23421920030080:rname="ucl",
   rmta="UK.AC.UCL.CS",rpass=" "
#
#
#-***********************
```

Figure 5.5: Example of X.400 Inbound Table

### 5.2.8 Testing Outbound Connections

- Go to the PP log directory.
- Look at the X.400 incoming log. The log should look like those in Figure 5.6.

```
2/ 7 15:41:38 x400in   22798 (#52     )  Starting x400in (via X400(84) inbound)
2/ 7 15:41:41 x400in   22798 (#52     )  Connection from
key='X121+5052334300018'   mta='manta.cng.dit.csiro.au'  password=' '
2/ 7 15:41:41 x400in   22798 (#52     )  No information found in the
X.400 incoming tables
2/ 7 16:41:41 x400in   22799 (#52     )  x400in/main error: rts_decode_request
2/ 7 16:41:38 x400in   22799 (#52     )  Starting x400in (via X400(84) inbound)
2/ 7 16:41:41 x400in   22799 (#52     )  Connection from
key='"PHL"/X121+302049600717'   mta='foo'  password='foobar'
2/ 7 16:41:41 x400in   22799 (#52     )  No information found in the
X.400 incoming tables
2/ 7 16:41:41 x400in   22799 (#52     )  x400in/main error: rts_decode_request
```

Figure 5.6: Example of X.400 Incoming Logs

From this log, the following remote site information may be obtained, which can then be inserted as two entries in the X.400 inbound and outbound tables.

- The TSAP: None for the first entry and "PHL" for the second.
- The Network Address: 5052334300018 for the first entry and 302049600717 for the second.
- The MTA Name: "manta.cng.dit.csiro.au" for the first entry and "foo" for the second.
- The password - " " for the first and "foobar" for the second.

However, note that:

- A remote site does not have to be specified in the X.400 inbound table for a connection to be successful. If in the *tailor* file the X.400 Inbound Channel is tailored with `ininfo=sloppy`, then all remote connections are accepted.
- If on the other hand MTA name and password validation is to be ignored, then for that remote entry in the X.400 inbound table, `other=sloppy` is added.

### 5.2.9 How to make a test X.400 connection to a remote site

- Go to the table directory.
- Put the remote site information in the appropriate tables.

- Use the channel in "ping" mode to make an RTS connection to the remote site. This is done by running the channel binary as follows:

  ```
  x400out88 ping -m mta -c channel
  ```

  Similarly for *x400out84*.

- Check the PP logs. The failure/success will be recorded there.

- The following failure reasons may be given:

  - "Validation failure": The remote site has either got an erroneous PP entry in their tables, or they haven't recorded your entry at all.
  - "Unacceptable dialogue mode": Either the remote site does not support monologue or the RTS ASN.1 encoding is bust.
  - "Connect request refused on this network connection": remote site is either down or its network listener is not running.
  - "Busy": remote site is busy, try again later.
  - "Protocol Error": protocol problems.
  - "Remote system problem": remote site's process has terminated abruptly.
  - "Timer expired": remote system's process is looping.

- If the "ping" was successful, then try transmitting a proper message, using the 'mail' program. If during transmission a protocol error is received then this needs to be resolved. To analyse this problem, it may be necessary to obtain the required X.400 information by turning up the rts, session, and transport logging in the PP *tailor* file. This can be done by adding the following 3 entries:

  isodelog rtsaplevel file=rts level=pdus

  isodelog ssaplevel file=session level=pdus

  isodelog tsaplevel file=transport level=pdus

### 5.2.10 A Common X.400 Incoming Connection Problem

Sometimes a remote site attempts to do an RTS recovery, due to some transmission failure. PP at the moment, does not support recovery, and so a "cannotRecover" reason is given. This sometimes does not terminate the connection due to a bug in the remote site's software, and so it keeps on trying, which leads to much network activity.

The X.400 Incoming logs will look something like:

```
2/ 8  8:30:31 x400in   01392 (#52     )
       Starting x400in (via X400(84) inbound)
2/ 8  8:30:33 x400in   01392 (#52     )
       RtBInit, [Congestion at RtSAP]
       rejecting attempted recovery
```

To get the network address of the remote site attempting the recovery, look at the ISODE tsapd.log, and then match the network address against the X.400 inbound table keys.

# Chapter 6

# Additional Capabilities

This chapter describes various units that may be added to PP to provide specific functionality. These units tend to provide new protocol functionality.

## 6.1 G3Fax capability in PP

This section describes PP's G3Fax capability and how to add it to the configured system. The G3Fax capability is divided into several components:

- Addressing
- Tables
- Conversion
- Transmission and Reception

These are described below.

### 6.1.1 Fax Addressing

To enable an address to contain the information required for fax delivery, PP recognises two specific X.400 domain defined attributes:

`FAX:` the value of this attribute is used to identify the remote fax machine. It can be the actual telephone number of the machine (e.g. `/FAX=+44 602 790278/`), in which case it is used directly. Or it can be a string which is looked up in the *fax* table to get the required information (e.g. `/FAX=xtel/`)

Note that the `FAX` attribute is compulsory. The PP fax transmission channel will bounce the message if this attribute is not present.

Note also that the number may contain symbols like '`+`' and '`P`'. The meaning and processing of these symbols is specific to the driver being used (see Section 6.1.4).

`ATTN:` the value of this attribute is a descriptive string. This string is printed on the coverpage of the fax along with the key `For the attention of`.

The `FAX` attribute identifies the remote fax machine and the `ATTN` attribute gives an informative string for presentation on the coverpage. Note that the presence of one or both of these attributes is not sufficient to route a message through PP for delivery via the fax transmission channel. Addresses will still have to be routed to the fax transmission channel via the conventional PP routing tables (the *channel* table and/or the the *users* table).

### 6.1.2 The Fax Table

This section describes the format of the table used by the fax related channels and filters. As the table is shared by both the conversion filters and the fax transmission channel, it is appropriate to describe the format and contents of the table here.

The general format of the table is the same as that of all PP tables. That is:

```
key:value
```

The table contains a mixture of delivery information and formating information. The main type of entry is of the form

```
key: number,description
```

Where

`key:` is the value of the domain defined attribute `FAX` described in Section 6.1.1.

`number:` is the telephone number of the remote fax machine.

Any special symbol processing and meaning given to special values is a specific to drivers and described in Section 6.1.4

`description:` is a string describing the remote site. This description, if given, is printed on the coverpage by *hdr2fax*. *hdr2fax* maps the sequence "`\n`" to a newline when printing the string.

The fax table may also contain several hardwired keys used by the conversion filters and the transmission channel. These are:

`keyfont:` the value is the name of a file containing the font used by *hdr2fax* when printing out keys on the coverpage.

`valuefont:` the value is the name of a file containing the font used by *hdr2fax* when printing out the values associated with the keys.

`textfont:` the value is the name of a file containing the font used by *ia52fax* for all text and by *hdr2fax* for any subsiduary information it puts on the coverpage.

`xstart:`

`ystart:` the values of these are used by *hdr2fax* to identify the top left hand corner of the coverpage. The values can be given in units, `cm` for centimetres or `in` for inches. If no unit is specified, the value is assumed to be in millimetres.

`tab:` the value is the minimum amount of space to leave between the start of the `key` and the start of the `value` on the coverpage. It can be specified with units in the same way as `xstart` and `ystart`.

`coverpage:` the value is the name of a file containing the bitmap of a design which is used by *hdr2fax* as a backdrop to the coverpage.

`localOrg:` the name and address of the local organisation. This is printed by *hdr2fax* on the coverpage. Note that `\n` is mapped to a newline.

`localNumber:` the telephone number of the local organisation. This is printed on the coverpage by *hdr2fax*.

`postscript:` supplementary information which is printed at the bottom of the coverpage by *hdr2fax*. Note that `\n` is mapped to a newline.

`int_prefix` the prefix required to phone international numbers.

Figure 6.1 shows an example fax table.

```
#format information
keyfont:        /fonts/bh-lucida-bold-r-normal-sans-26
valuefont:      /fonts/bh-lucida-medium-i-normal-sans-26
textfont:       /fonts/bh-lucida-medium-r-normal-sans-26
xstart:         1.5in
ystart:         2.5in
tab:            2.5in
coverpage:      /bitmaps/xtel.bm
int_prefix:     010
localOrg:       X-Tel Services Ltd,\nUniversity of Nottingham,\nUniversity Park,\nNottingham,\nNG7 2RD
localNumber:    +44 602 790278
postscript:     This fax was automatically generated by PP software.
#recipient information
lancaster.xtel.co.uk:       0,A local print
nott:           783450,Computer Science,\nNottingham University
cripps:         588138,Cripps Computer Centre,\nNottingham University
ucl:            071 387 1397,UCL,\nGower Street,\nLondon WC1E 6BT
bobbies:        999,The Police Station
```

Figure 6.1: Example of Fax Table

### 6.1.3 Fax Conversion

This section describes the fax conversion facilities supplied with PP. There are two filters:

*hdr2fax*: which converts an RFC-822 header to a G3Fax bodypart containing a coverpage.

*ia52fax*: which converts an ia5 bodypart to a G3Fax bodypart with multiple pages.

Note that the size of the page produced is driven by the `FAXPAGESIZE` macro from *Make.defs*. If this macro is not defined, the filters will produce US size pages. That is 216 mm wide by 273 mm high. If `FAXPAGESIZE` equals `-DA4`, the filters will produce A4 size pages, 210mm wide by 297 mm high.

These filters use fonts and bitmaps encoded in PP's own format. There are two subsiduary tools which can be used to produce the required format of fonts and bitmaps:

`xfontconv`: this converts an X windows' font into PP's format. It is run as follows:

```
xfontconv -font fontName -file fileName
```

where `fontName` is the name of the required font which must be present in the local X server and `fileName` is the name of the file in which to place the PP encoded version of the font.

`ximageconv`: this converts an X windows' bitmap into PP's format. It is run as follows:

```
ximageconv -in xbitmapFile -out fileName
```

where `xbitmapFile` is the file containing the bitmap to convert and `fileName` is the file in which to place the PP encoded version of the bitmap.

#### The *hdr2fax* Filter

The *hdr2fax* filter is used in conjunction with the general filter controller *fcontrol*. It produces a coverpage for a message, based on the information it is passed via the tailored `outinfo` field and the information it can glean from the RFC-822 header field.

The coverpage is built up as a series of key/value pairs separated by a set horizontal distance (`tab` see Section 6.1.2). These key/value pairs include To/value, From/value etc.

The information passed via the `outinfo` field is standard and as such the `outinfo` field should be tailored thus

```
hdr2fax -table $(table) -from $(400sender) -to $(400recip) \
-subject $(ua-id) -outmta $(outmta)
```

Although the `outinfo` is standard, you may omit or alter components of the field if you wish. This will result in the corresponding key/value pair being omitted from the coverpage. The only component which is compulsory is the `-table` component. Note that the `$(...)` values will be expanded automatically by *fcontrol*.

The components of the `outinfo` field are:

`-table:` the name of the fax table (see Section 6.1.2)

`-from:` the originator of the message.

`-to:` the recipient of the message.

`-subject:` the subject of the message.

`-outmta:` the pseudo mta to which this message is to be delivered.

There is a further consideration concerning the tailoring of *hdr2fax*. *hdr2fax* produces a coverpage specific to one recipient of a message. Therefore to ensure that each recipient gets a coverpage specific to themself, the tailoring for the *hdr2fax* should have the variable `solo-proc` set to `yes`.

Figure 6.2 shows an example of how *hdr2fax* should be tailored.

```
chan    822tofax        prog=fcontrol,type=shaper,
  hdrin=822-us,hdrout=g3fax,
  outinfo="hdr2fax -table $(outtable) -from $(400sender)
      -to $(400recip) -subject $(ua-id) -outmta $(outmta)",
  outtable=fax, cost=20,solo-proc=yes,
  show="hdr.822 -> g3fax mapper"

chan    ia5tofax        prog=fcontrol,type=shaper,
  bptin=ia5,bptout=g3fax,
  outinfo="ia52fax -table $(outtable)",outtable=fax,                    10
  show="ia5 -> g3fax mapper"

chan    fax             prog=G3Fax, show="Fax channel",type=both,
  content-in=p22, maxproc=1,
  hdrout=g3fax, bptout=g3fax, drchan=dr2rfc,
  outtable=fax, sort="user time",
  outinfo="out=/dev/faxout,confirm=always,softcar=used",
  ininfo="in=/dev/faxin,softcar=used,master=faxmaster@xtel.co.uk"
```

Figure 6.2: Example Tailoring of Fax Channels

### The *ia52fax* Filter

The *ia52fax* filter is used, in conjunction with the general filter controller *fcontrol*, to convert ia5 bodyparts into multiple page g3fax bodyparts.

The `outinfo` field needs to specify which font the filter should use. This can be

- either indirectly via the `textfont` specified in the fax table (see Section 6.1.2). In which case the `outinfo` field should contain the component `-table $(table)`.
- or directly by specifying the name of the file containing the font to be used. The `outinfo` field should contain the component `-font filename`

Figure 6.2 shows an example of how *ia52fax* should be tailored.

### 6.1.4 The Fax Transmission and Reception Channels

This section describes the fax transmission and reception facilities supplied with PP. This code is divided into PP skeletons for an outbound fax channel and an inbound fax daemon, and drivers that are written to use these skeletons. At present there are only two drivers written using these skeletons. These drivers are for the following systems:

- the Panasonic Systemfax 250
- the Fujitsu dexNet200

At the time of writing the inbound daemon is still in the beta stage.

The inbound daemon receives an incoming fax, wraps it up into an x400 message and sends it to a specified address. The outbound channel transmits a message to the remote fax machine via the local fax machine.

With channel pairing, both these entities can be tailored as one channel. Figure 6.2 shows an example of how these could be tailored. This example is written for the Panasonic Systemfax 250 driver.

Before describing the details germane to the individual drivers, the details germane to the PP skeletons are described. The key data structure linking the skeletons and the drivers is the structure `FaxCtlr`. The elements of this structure are mostly function pointers which should be filled in by the driver code. The structure contains other elements which mostly relate to the inbound daemon. These include: the address to send inbound faxes to; the subject to attach to the message sent; and the channel to run as.

In terms of tailoring, there is only one PP specific tailor element. This is for the outbound channel and is the element `confirm`. If this element is set to `always`, a positive delivery report will be sent back to the originator of the message when the message has been successfully transmitted. If set to `never`, positive delivery reports aren't sent. The default action is not to send delivery reports.

#### The Panasonic Systemfax 250 Driver

#### The Outbound Transmission Channel

The Panasonic Systemfax's outbound channel's behaviour can be altered by several key/value pairs in the `outinfo` tailor field:

`out=device:` the outbound device on which to contact the local fax machine. By default it will try to use `/dev/faxout`